REMARKS:

The applicant would like to thank the Examiner for the thorough review and consideration of the present application.

Independent claim 1, 71, 139 and 140 have been amended to recite the limitations more clearly. No new matter has been added. For example, applicant has amended claim 1, 71, 139 and 140 to recite the limitation that "**model of said software includes input and output behavior of said software under control of said software controller; wherein said input behavior of said software includes at least one control enabling structure, wherein said control enabling structure interacts with said software controller; wherein said output behavior of said software includes at least one display screen region**". The recitation is supported at least at paragraphs [29, 36, 131, 132, 133, 134, 136, 137, 138, 139, 140, 163, 167, 168, 169, 180, 181, 321, 347, 349]. Also to distinguish this invention from prior art, applicant recites the limitation for running the software modeling and software simulation process "**automatically and programmatically**". The recitation is supported at least at paragraphs [38, 159, 179, 185, 186].

Claim 1-140 are currently pending.

Claims 1-3, 48-51, 71-73, 116-119, and 139-140 stand rejected under 35 U.S.C.§102(e) as allegedly being anticipated by Chailleux (U.S. Patent Application Publication No. 2002/0109736 Al).

With respect, it is submitted that there are fundamental differences between the rejected claims and the teachings of Chailleux.

Chailleux discloses at [ABST] "a system for producing presentations of computer application programs". It is well known in the art that in order to produce presentations of the software, the software needs to be operated or controlled interactively. One of fundamental differences between the rejected claims and Chailleux teaching lies in how to interact with the software, in other words, how to operate the software by feeding interactive actions into the software or "computer application programs", and sequences of which are to be captured and presented to end users.

Chailleux discloses two different types of author actions or manual operations, at paragraph [69], "The sequence **author is operating the calculator interface** and can **depress the 'Pause' key** at any time in order to capture a screenshot into the Leelou file".

The first type of author actions as disclosed by Chailleux at paragraph [69],

"the sequence **author is operating the calculator interface**", is the author actions feeding directly into the software user interface. For example, it is well known in the art that in order to present end users how to input a digit into calculator software, the author needs to operate the calculator interface by clicking a keypad interface, which is depicted in FIG. 5 of Chailleux, or press a numeric key from computer keyboard manually. This type of **manual interactions** between the software and the author is required to operate the software in Chailleux disclosure. The actions by the author are targeted at and consumed by the software. For simple, we might term this type of author actions as **software interaction**.

The second type of author action is disclosed by Chailleux at paragraph [69], "depress the 'Pause' key at any time in order to capture a screenshot into the Leelou file", and at paragraph [60], "human operator to select screen shots, designate cursor movements, define 'bubble' text, etc., in order to create a sequence. Other embodiments may automate the process where, for example, another computer program automatically performs one or more of the **authoring steps** described herein". Obviously, these author actions are targeted at and consumed by an authoring program, as disclosed at paragraph [46], "an authoring program, Leelou, for use by an author to create a sequence of slides along with animations, text and other features for playback at a later -time". For simple, we might term this type of author actions as **authoring interaction**. The **authoring interaction** has nothing to do with the software or operating the software, and the authoring program does not interact with nor control the running software.

Furthermore, the purpose of two types of author actions is different. The first type of author actions as a sequence of **software interaction** is intended to be captured in the form of slides as "subject matter" as disclosed by Chailleux, and that captured sequence is to be presented or replayed back to end users. In contrast, the second type of author actions as **authoring interaction**, such as "depress the 'Pause' key', designate cursor movements, define 'bubble' text, etc." is out of the software running context and its sequence is never intended to be captured, nor presented or replayed back to end users.

Based on the purpose and target of the interactions, and its intended end users, s**oftware interaction** and **authoring interaction** are two distinctly different interactions. Chailleux might suggest to automate the **authoring interaction**, the interaction between the author and the authoring program by a computer program, but Chailleux does not teach nor suggest to automate the **software interaction**, the interaction between the author and the software by a computer program. However, automating the **software interaction** by a computer program without the author operating the software, is one of inventive mechanisms disclosed and claimed in this application.

In the specification of this application, applicant discloses an embodiment

for automating the **software interaction** by a **software controller**, a mechanism to interact with the software by the software controller automatically and programmatically instead of a human author manually. For example, the applicant discloses, at paragraph [26], "the present invention models the interactive software S running on digital computers as a physical device with causality. It develops the process of interaction between the **software** and its **human user** in a **plant-controller servo-mechanism** similar to the processes that system science and engineering apply to physical processes. A **software controller** is developed to **model** user intelligence. A software dynamic system is invented to **model** and **simulate** the **software**, **software controller** and their **interactions**", and further at paragraph [28], "The **software controller** is a **programmable agent A** encapsulating the strategies to perform tasks in a software space", and further, applicant explicitly teaches at paragraph [136], "The command I as the input to the software S is computed programmatically by the **agent A** 226 in real-time and fed into the software S 221 through the input terminal 220. The actions such as the mouse input 222 and the key input 223 are external to and independent of the underlying software S 221".

Furthermore, applicant discloses another inventive mechanism for identifying the model of the software through the automated **software interaction** under the control of the software controller. For example, applicant discloses, at paragraph [29], "The software S is connected with the **agent A** in a closed-loop fashion as a software dynamic system, in which the running software S is exerted by external commands that are computed by the **agent A** while its input/output behavior is controlled and observed in real-time in order to identify a model of the software, $\bar{\bar{S}}$. The modeled $\bar{\bar{S}}$ is a system construct that preserves the state transition mechanism and the control enabling structure of the software S when under control of the **agent A**", and further, applicant explicitly teaches at paragraph [186], "The online modeling process is fully mechanized without any intervention from the outside since the agent is modeled exactly to carry out all the interactions against the software autonomously. The sole purpose of the modeling is to identify the model $\bar{\bar{S}}$(S) 315 that powers $\bar{\bar{S}}$ under control of the same agent A 311. Once the on-line modeling is performed, and $\bar{\bar{S}}$(S) is developed, the real software S 310 can be replaced by the acquired model $\bar{\bar{S}}$ 315 in further development".

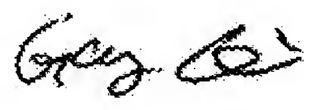
Furthermore, applicant discloses another inventive mechanism for simulating the automated **software interaction** by utilizing the **same software controller** to control the identified model of the software, for example, applicant discloses, at paragraph [30], "The modeled software $\bar{\bar{S}}$ is connected with the

same agent A again in the same closed-loop fashion as a simulated software dynamic system, in which the modeled software S is exerted by the same commands computed by the **agent A** in order to simulate the interaction between the software and the agent without the real software S presence", and further, applicant explicitly teaches at paragraph [314], "Controlling the model of the software S by the **same agent A** again in the simulation itself embodies and preserves the structure and model knowledge about the underlying software S. That dynamic knowledge can be reconstructed within the execution flow of the agent in the simulation. For example, there may be tens of thousands of sampled states in the biped animation, of which 100 states are controlled and simulated by the APIProbe. Each state is intrinsically modeled with the dynamic information that is related to the biped animation. For example, between the 25th state and the 26th state, there is a mode transition from the footstep to the freeform animation predictably. Each distinct state is programmatically accessible state-by-state for any further development".

Since Chailleux does not disclose, teach nor suggest to automate the **software interaction**, let alone **modeling**, **identifying** and **simulating** the **software interaction** as recited in applicant's independent claims 1, 71, 139, and 140 that have been amended to more clearly characterize and distinguish applicant's invention from prior art, therefore, claims 1, 71, 139 and 140 are submitted to be novel, inventive and patentable. Since dependent claims 2-3, 48-51 are depend upon claim 1, dependent claims 72-73 and 116-119 are depend upon claim 71, applicant submits that claim 2-3, 48-51, 72-73 and 116-119 are patentable at least by virtue of their dependency from their base claims.

Therefore, Applicant respectfully requests that the rejection of claims 1-3, 48-51, 71-73, 116-119, 139-140 under 35 U.S.C.§102(e) be reconsidered and withdrawn, and all the claims are in condition for allowance.

Respectfully submitted,



By: Dr. Gang Qiu (Inventor)

Date: March. 6, 2008

20910 Pepper Tree Ln,
Cupertino, CA 95014
gqiu8@yahoo.com